Hepatoprotective effects of systemic ER activation

Spheroid RNA-seq - Downstream analysis

Christian Sommerauer & Carlos Gallardo

25 July, 2023

```
# source and library import
source('code/00_helper_functions.R')
library(clusterProfiler)
library(tidyverse)
library(org.Hs.eg.db)
library(Mfuzz)
library(biomaRt)
library(GenomicRanges)
library(ComplexHeatmap)
```

```
# color palettes
colPals <- list()
colPals$RdBu <- rev(RColorBrewer::brewer.pal(n=11, name = 'RdBu'))
colPals$UpDown <- setNames(colPals$RdBu[c(10,2)],
                           c('up', 'down'))
```

Load data

```
# spheroid TPM-normalized expression
spheroid_tpm <- read.table(
  file = 'results/spheroid_TPM_norm_counts.txt',
  sep = '\t',
  header = T,
  quote = ''
)
# spherioid differential expression results
inhibitor_RDS <- readRDS("results/Spheroid_inhibitors_DEGlists.rds")

universe <- inhibitor_RDS$unfilt$TEADap_vs_control %>% pull("external_gene_name")

TEADap_inh_vs_control <- inhibitor_RDS$filt$TEADap_vs_control
TEADsf_inh_vs_control <- inhibitor_RDS$filt$TEADsf_vs_control

# human gene promoter regions (ensembl 109)
promoter_regions <- read.table(
  file = "data/ensembl_hsap_feb2023_promoter_regions.bed",
  sep = "\t",
  quote = "",
  header = F)
colnames(promoter_regions) <- c(
  'chromosome', 'promoter_start', 'promoter_end', 'ensembl_gene_id',
  'external_gene_name', 'ensembl_transcript_id', 'strand', 'gene_biotype', 'description')

# TEAD1 binding sites geneome-wide (HOCOMOCOv11 + PWMScan)
tead1_tfbs <- read.table(
  file = "data/TEAD1_tfbs_hg38_hocomoco_v11_pwmscan.bed",
  sep = "\t",
  quote = "",
  header = F)
colnames(tead1_tfbs) <- c('chromosome', 'start', 'end', 'sequence', 'score', 'strand')

# mouse-human orthologs (ensembl 105)
mouse_human_orthologs <- read.table(
  file = 'data/ensembl_mmus_hsap_dec2021_orthologs.tsv',
  sep = '\t',
  header = TRUE,
  quote = '')
```

```r
# mouse differentially expressed genes
DEGs_mouse <- readRDS('results/bulkRNAseq_mmus_DEGs.rds')
```

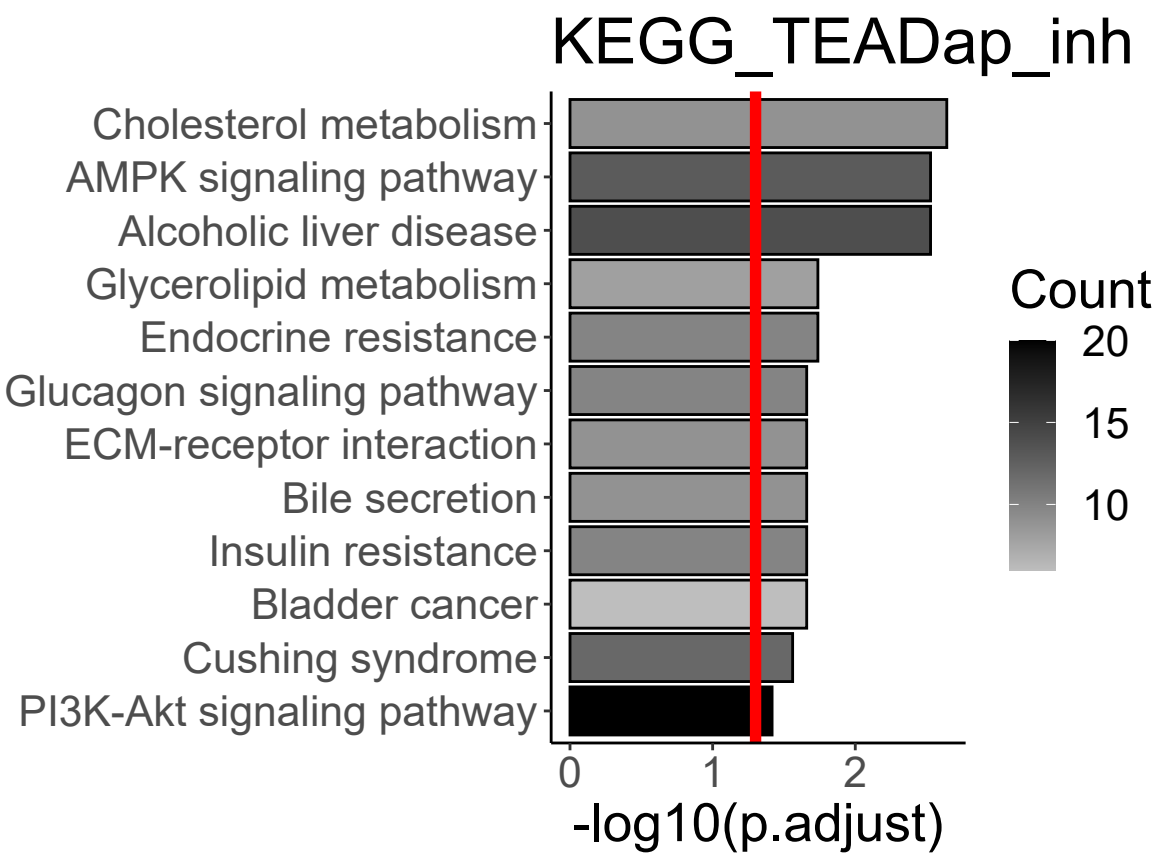
KEGG over-representation analysis

```r
TEADap_inh_vs_control_entrez <- bitr(TEADap_inh_vs_control$ensembl_gene_id, fromType = "ENSEMBL", toType = "ENTREZID", OrgDb = "org.Hs.eg.db")
TEADsf_inh_vs_control_entrez <- bitr(TEADsf_inh_vs_control$ensembl_gene_id, fromType = "ENSEMBL", toType = "ENTREZID", OrgDb = "org.Hs.eg.db")

KEGG_TEADap_inh <- enrichKEGG(gene = TEADap_inh_vs_control_entrez$ENTREZID,
          organism = "hsa",
          keyType = "kegg",
          pAdjustMethod = "BH",
          qvalueCutoff = 0.05)
KEGG_TEADap_inh <- as.data.frame(KEGG_TEADap_inh@result)
#write.table(KEGG_TEADap_inh, "results/KEGG_TEADap_vs_control_FigS8A.txt", sep="\t", quote=F, row.names = F)

KEGG_TEADsf_inh <- enrichKEGG(gene = TEADsf_inh_vs_control_entrez$ENTREZID,
          organism = "hsa",
          keyType = "kegg",
          pAdjustMethod = "BH",
          qvalueCutoff = 0.05)
KEGG_TEADsf_inh <- as.data.frame(KEGG_TEADsf_inh@result)


#This function tells you the gene symbols in each KEGG pathway (changed by TEADap)
gimme_SYMBOL_from_KEGG <- function(table, pathway) {
 table_subset <- table %>%  filter(Description==pathway) %>% pull("geneID")
 split_entrez <- unlist(strsplit(as.character(table_subset), "/"))
 symbols <- bitr(split_entrez, fromType = "ENTREZID", toType = "SYMBOL", OrgDb = "org.Hs.eg.db")
 symbols <- symbols %>%  mutate(Pathway = pathway)
 symbols

}

# This code exports the genes inside the top12 KEGG pathways of TEADap vs control with the corresponding human-readable symbols.
KEGG_top12_pathway_geneList <- list()
for (i in KEGG_TEADap_inh$Description[1:12]) {
  KEGG_top12_pathway_geneList[[i]] <- gimme_SYMBOL_from_KEGG(table=KEGG_TEADap_inh, pathway=i)
}

#saveRDS(KEGG_top12_pathway_geneList, "results/top12KEGGpathways_gene_content.rds")
```

Plotting the KEGG pathways

```r
# Import the KEGG table (run in June 2023, KEGG version from April)
KEGG_TEADap_inh <- read.delim("results/KEGG_TEADap_vs_control_FigS8A.txt")

# Note: This following section is looped, so one can technically plot many results at once.
list_KEGG <- list(KEGG_TEADap_inh=KEGG_TEADap_inh)

combined_genesets <- list()

for (i in names(list_KEGG)) {

combined_genesets[[i]] <- list_KEGG[[i]] %>%
               dplyr::arrange(p.adjust) %>%
               dplyr::slice(1:12)

names(combined_genesets[[i]])
order <- list()
order[[i]] <- combined_genesets[[i]] %>% dplyr::pull(Description)
# Plot and order by p.adjust
ggplot_bar <- ggplot(combined_genesets[[i]], aes(x=factor(Description, levels=order[[i]]), y=-log10(p.adjust), fill=Count)) +
  geom_col(color="black") +
  coord_flip() +
  scale_x_discrete(limits=rev) +
  #scale_y_continuous("-log10") +
  scale_fill_gradient(high = "black", low="grey") +
  theme_classic() +
  theme(text=element_text(size=20)) +
  geom_hline(yintercept = -log10(0.05), color="red", linewidth=2) +
  xlab("") +
  ggtitle(i)

date <- gsub("-", "", Sys.Date())
```

```
print(ggplot_bar)

#ggsave(filename = paste0("results/","SupplFig8A_KEGG_RNAseq_top12_", i, ".pdf"), width=10, height=7)

}
```



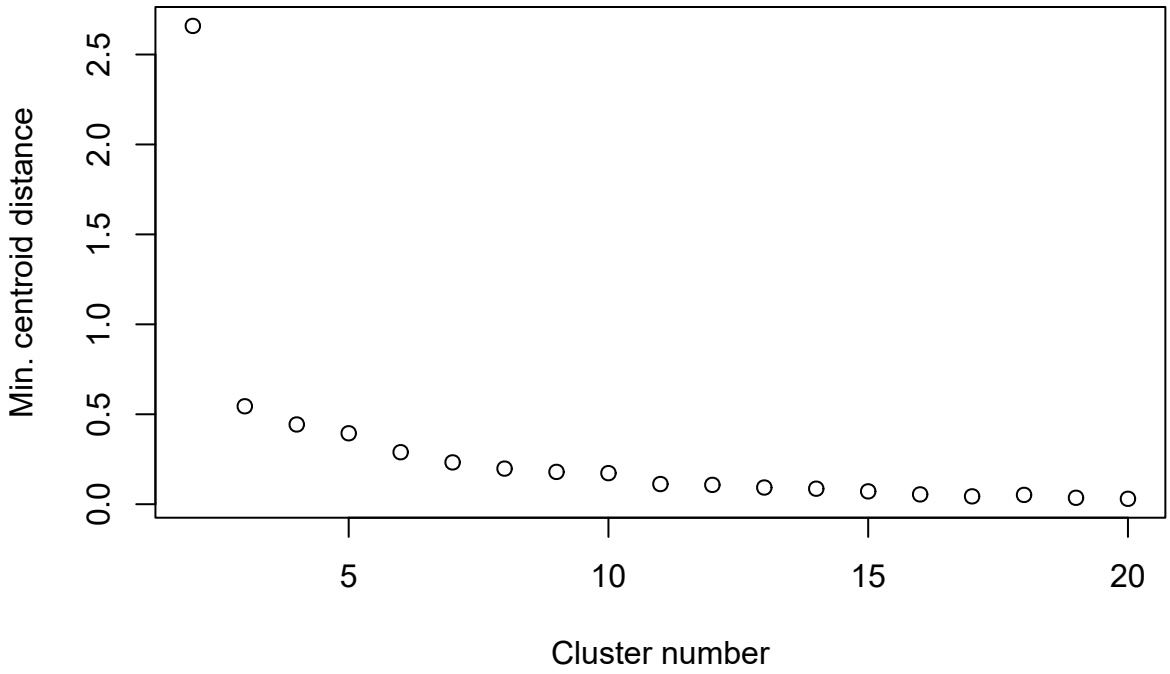
Gene expression trend analysis

```
# CLUSTERING WITH ONLY Inhibitors DEG

# Clustering of expression profiles
DESeq2_DEGs_clust <- list(TEADap_inh_vs_control, TEADsf_inh_vs_control)

names(DESeq2_DEGs_clust) <- c("TEADap_inh_vs_control", "TEADsf_inh_vs_control")

DEGs_union <- lapply(DESeq2_DEGs_clust, function(x) x$ensembl_gene_id) %>%
  unlist() %>%
  unique()

# TPM - normalized table with the average TPM (computed in Spheroid DESeq2 analysis script)
nonNorm_Counts <- read.delim("results/spheroid_TPM_norm_counts_mean.txt")

TPM_Normalized <- nonNorm_Counts %>% tibble::column_to_rownames("ensembl_gene_id") %>%
  dplyr::select(-description, -external_gene_name)
colSums(TPM_Normalized)
```

```
##  control   TEADap   TEADsf
## 999994.4 999992.2 999990.9
```

```
eset <- TPM_Normalized %>%
  dplyr::filter(row.names(.) %in% DEGs_union)

# zscore data (mean=0, sd=1)
eset <- new('ExpressionSet', exprs = as.matrix(eset)) %>%
  Mfuzz::standardise()
```

```
# estimate fuzzifier parameter for clustering
m_eset <- Mfuzz::mestimate(eset)

# determine cluster number with minimum centroid distance
Mfuzz::Dmin(eset, m = m_eset, crange = seq(2,20,1), repeats = 5)
```




```
##  [1] 2.65902958 0.54441975 0.44330534 0.39431485 0.28913332 0.23245431
##  [7] 0.19791434 0.17962183 0.17273033 0.11209072 0.10749572 0.09276878
## [13] 0.08623862 0.07135137 0.05453264 0.04367029 0.05169969 0.03569742
## [19] 0.03001923
```

```
set.seed(3452)

# generate mfuzz clusters (n=4)
clusters <- mfuzz(eset, c = 2, m = m_eset)

# check correlation of cluster centroids
cor(t(clusters[[1]]))
```

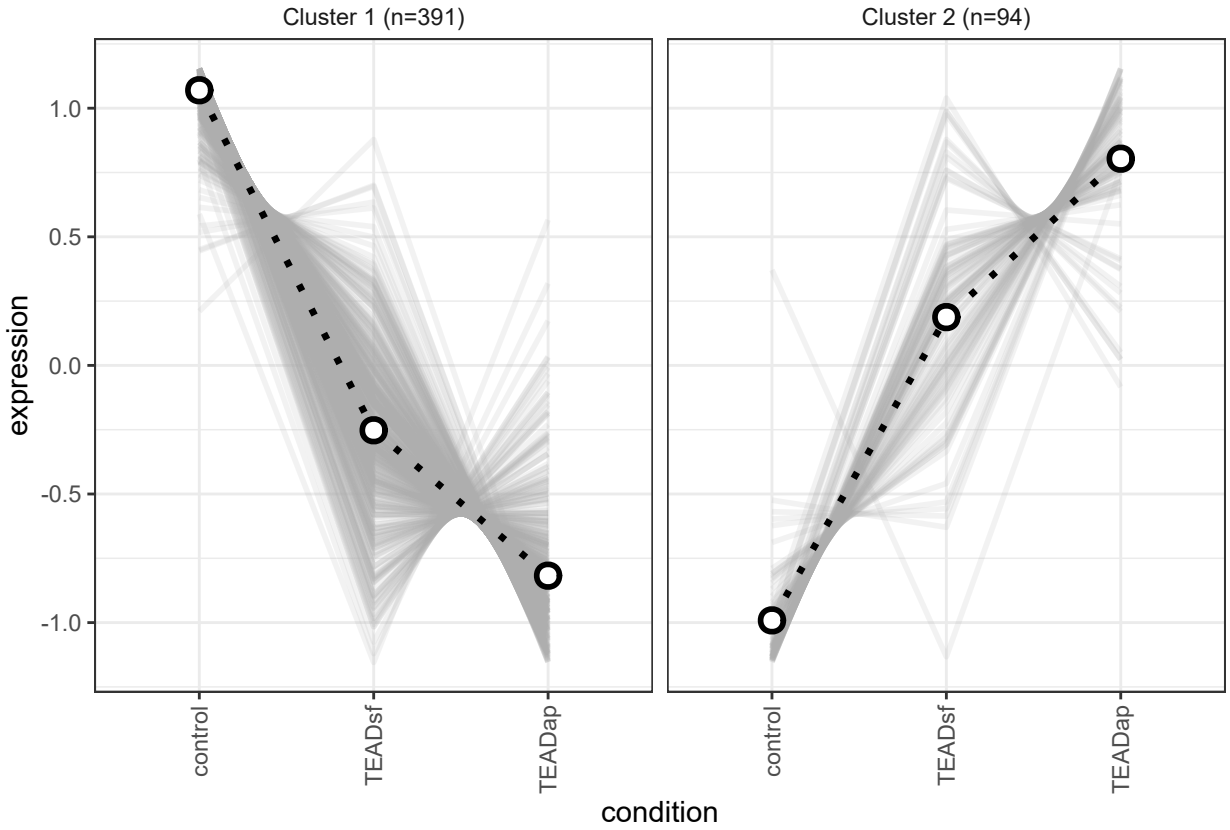
```
##           1          2
## 1  1.0000000 -0.9988336
## 2 -0.9988336  1.0000000
```

```
# get cluster membership values of genes
cluster_memberships <- acore(eset, cl = clusters, min.acore = 0.0)

# assign to cluster with top membership value
cluster_memberships <- do.call(rbind,
                               lapply(seq_along(cluster_memberships),
                                      function(x) {data.frame(CLUSTER=x,
                                                              cluster_memberships[[x]])})) %>%
  dplyr::mutate(CLUSTER=dplyr::recode(CLUSTER, !!!setNames(c(2,1), seq(1,2,1))))

# check number of genes per cluster
cluster_gene_counts <- table(cluster_memberships$CLUSTER)
cluster_gene_counts
```

```
##
##   1   2
```

```
## 391  94
```

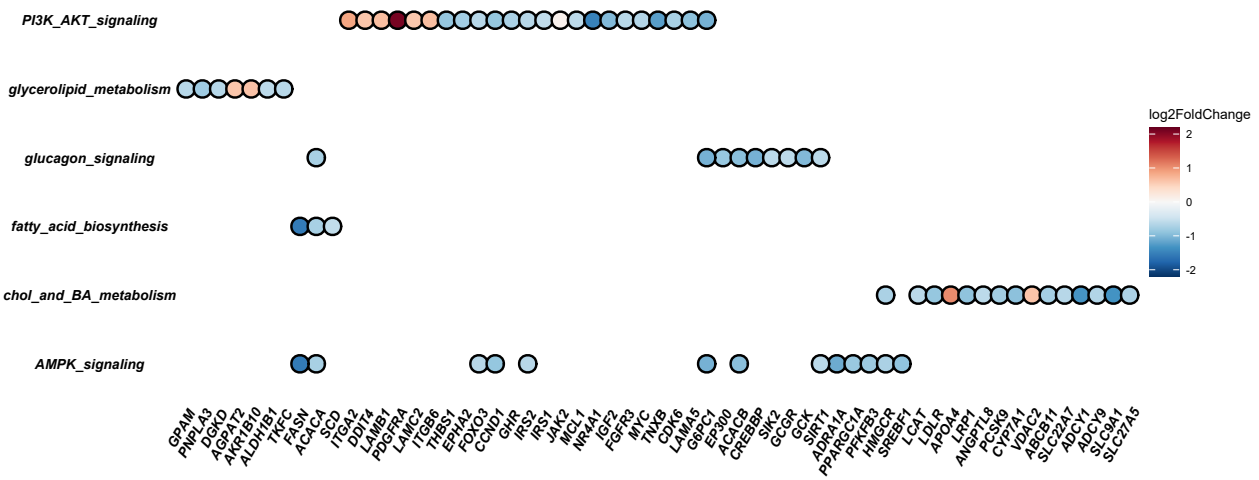
```
DEG_clusters <- list()
# extract gene profiles and cluster assignments
DEG_clusters$genes <- as.data.frame(exprs(eset)) %>%
  tibble::rownames_to_column(var = 'geneID') %>%
  tibble::add_column(GeneSymbol = .$geneID, .after = 'geneID') %>%
  dplyr::mutate(GeneSymbol=dplyr::recode(GeneSymbol,
                                         !!!setNames(inhibitor_RDS$unfilt$TEADap_vs_control$external_gene_name,
                                                     inhibitor_RDS$unfilt$TEADap_vs_control$ensembl_gene_id))) %>%
  merge(cluster_memberships, by.x = 'geneID', by.y = 'NAME', sort = F)

# extract cluster centroid profiles
DEG_clusters$centroids <- as.data.frame(clusters$centers) %>%
  tibble::add_column(geneID = paste0('centroid_', c(2,1)), .before = 'control') %>%
  tibble::add_column(GeneSymbol = paste0('centroid_', c(2,1)), .before = 'control') %>%
  dplyr::mutate(CLUSTER=c(2,1),
                MEM.SHIP=1) %>%
  arrange(CLUSTER)

df <- DEG_clusters$genes %>%
  dplyr::bind_rows(DEG_clusters$centroids) %>%
  tidyr::pivot_longer(cols = c("control", "TEADsf", "TEADap"),
                      names_to = 'condition',
                      values_to = 'expression') %>%
  dplyr::mutate(CLUSTER=factor(CLUSTER, levels = 1:2),
                condition=factor(condition, levels = c("control", "TEADsf", "TEADap")))

cluster_names <- c('1' = paste0("Cluster 1 (n=",cluster_gene_counts[1],")"),
                   '2' = paste0("Cluster 2 (n=",cluster_gene_counts[2],")"))

ggplot(df, aes(x=condition, y=expression, color=CLUSTER, group=geneID)) +
  geom_line(data = subset(df, !grepl('centroid', GeneSymbol)), size = 1,  color=alpha('#AEAEAE', 0.15)) +
  geom_line(data = subset(df, grepl('centroid', GeneSymbol)), size = 1.2, linetype="dotted", color="black") +
  geom_point(data = subset(df, grepl('centroid', GeneSymbol)), shape=21, size=3, stroke=1.5, fill='white', color="black") +
  #scale_color_manual(values = colPals$clusters) +
  facet_wrap(~CLUSTER, nrow = 1, labeller = as_labeller(cluster_names)) +
  theme_bw() +
  theme(strip.background = element_blank(),
        axis.text.x=element_text(angle=90, hjust=0.95, vjust=0.5))
```




```
#ggsave("cMeans_clustering_inhibitors_2clusters.pdf", width = 23, height=10, units="cm")
#write.table(DEG_clusters$genes, "cMeans_clustering_inhibitors_2clusters.txt", sep="\t", row.names = F, quote = F)
```

Gene expression fold change in key pathways (TEADap vs control)

```
pathway_genes <- list(
  glycerolipid_metabolism = c('GPAM','PNPLA3','DGKD','AGPAT2','AKR1B10','ALDH1B1','TKFC'),
  fatty_acid_biosynthesis = c('FASN','ACACA','SCD'),
  PI3K_AKT_signaling = c('ITGA2','DDIT4','LAMB1','PDGFRA','LAMC2','ITGB6','THBS1',
                         'EPHA2','FOXO3','CCND1','GHR','IRS2','IRS1','JAK2','MCL1',
                         'NR4A1','IGF2','FGFR3','MYC','TNXB','CDK6','LAMA5','G6PC1'),
  glucagon_signaling = c('EP300','ACACA','ACACB','CREBBP','SIK2','GCGR','G6PC1','GCK','SIRT1'),
  AMPK_signaling = c('FOXO3','ADRA1A','ACACA','PPARGC1A','FASN','ACACB','G6PC1',
                     'PFKFB3','HMGCR','IRS2','SIRT1','CCND1','SREBF1'),
  chol_and_BA_metabolism = c('HMGCR','LCAT','LDLR','APOA4','LRP1','ANGPTL8','PCSK9','CYP7A1',
                                          'VDAC2','ABCB11','SLC22A7','ADCY1','ADCY9','SLC9A1','SLC27A5')
)

df <- lapply(names(pathway_genes), function(x){
  inhibitor_RDS$unfilt$TEADap_vs_control %>%
    filter(external_gene_name %in% pathway_genes[[x]]) %>%
    mutate(pathway = x,
           external_gene_name = factor(external_gene_name, levels=pathway_genes[[x]])) %>%
    dplyr::select(pathway, external_gene_name, log2FoldChange) %>%
    arrange(external_gene_name)
}) %>% bind_rows()

ggplot(df, aes(x=external_gene_name, y=pathway, fill=log2FoldChange)) +
  geom_point(shape=21, size=4, stroke=1) +
  scale_fill_gradientn(colours = colPals$RdBu, limits=c(-2.2,2.2)) +
  theme_void(base_size = 8) +
  theme(axis.text = element_text(face = 'bold.italic'),
        axis.text.x=element_text(angle=60, hjust=0.9, vjust=0.9))
```



Promoter binding sites

```
# promoter regions
promoters_gr <- with(promoter_regions, GRanges(chromosome, IRanges(promoter_start, promoter_end), strand = strand))

# binding site regions
tead1_tfbs_gr <- with(tead1_tfbs, GRanges(chromosome, IRanges(start, end), strand = strand))

# overlap analysis ignoring strand information
overlap <- countOverlaps(promoters_gr, tead1_tfbs_gr, ignore.strand = TRUE)

# add the counts to the promoter data frame (promoter with max. binding sites per gene selected)
promoter_tead1_overlap <- promoter_regions %>%
  mutate(tead1_bs = overlap) %>%
  group_by(ensembl_gene_id) %>%
  slice_max(tead1_bs, with_ties = F)

# separate promoters by TEADap inhibitor DEGs and non-DEGs
df <- promoter_tead1_overlap %>%
  dplyr::select(ensembl_gene_id, external_gene_name, gene_biotype, description, tead1_bs) %>%
  mutate(type = ifelse(ensembl_gene_id %in% TEADap_inh_vs_control$ensembl_gene_id, 'DEGs', 'Non-DEGs'))

# get random background sets for permutation test
# set.seed(23)
non_DEG_set <- df %>% dplyr::filter(type=='Non-DEGs') %>% pull(ensembl_gene_id) %>% unique()
DEG_set <- df %>% dplyr::filter(type=='DEGs') %>% pull(ensembl_gene_id) %>% unique()
# random_genes <- lapply(seq(1,1000), function(x) sample(x = non_DEG_set, size = length(DEG_set)))
# saveRDS(random_genes, file = 'results/spheroid_TEADap_random_promoter_sets.rds')
random_genes <- readRDS(file = 'results/spheroid_TEADap_random_promoter_sets.rds')

# TEAD1 binding sites in DEGs vs non-DEG sets
non_DEG_bs <- lapply(random_genes, function(x){
  df %>%
    dplyr::filter(ensembl_gene_id %in% x) %>%
    pull(tead1_bs) %>%
    mean()
}) %>% unlist() %>% as.data.frame() %>% rename('.' = 'avg_bs')

DEG_bs <- df %>%
    dplyr::filter(ensembl_gene_id %in% DEG_set) %>%
    pull(tead1_bs) %>%
    mean()

print('Average binding sites in non-DEG sets:')
```

```
## [1] "Average binding sites in non-DEG sets:"
```

```
summary(non_DEG_bs)
```

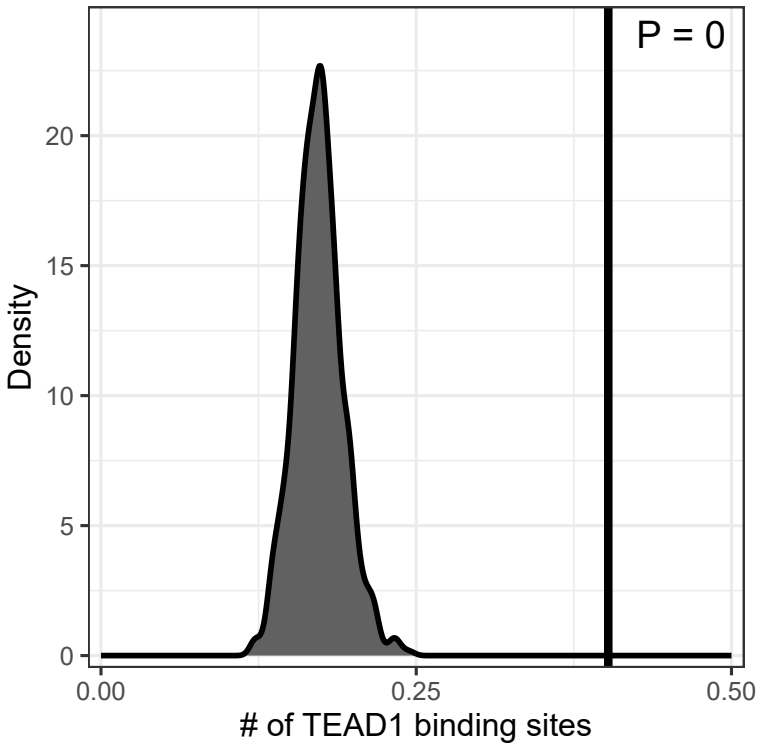
```
##      avg_bs
##  Min.   :0.1195
##  1st Qu.:0.1609
##  Median :0.1724
##  Mean   :0.1727
```

```
##  3rd Qu.:0.1839
##  Max.   :0.2460
```

```
paste('Average binding sites in TEADap inhibitor DEG set:', DEG_bs)
```

```
## [1] "Average binding sites in TEADap inhibitor DEG set: 0.402298850574713"
```

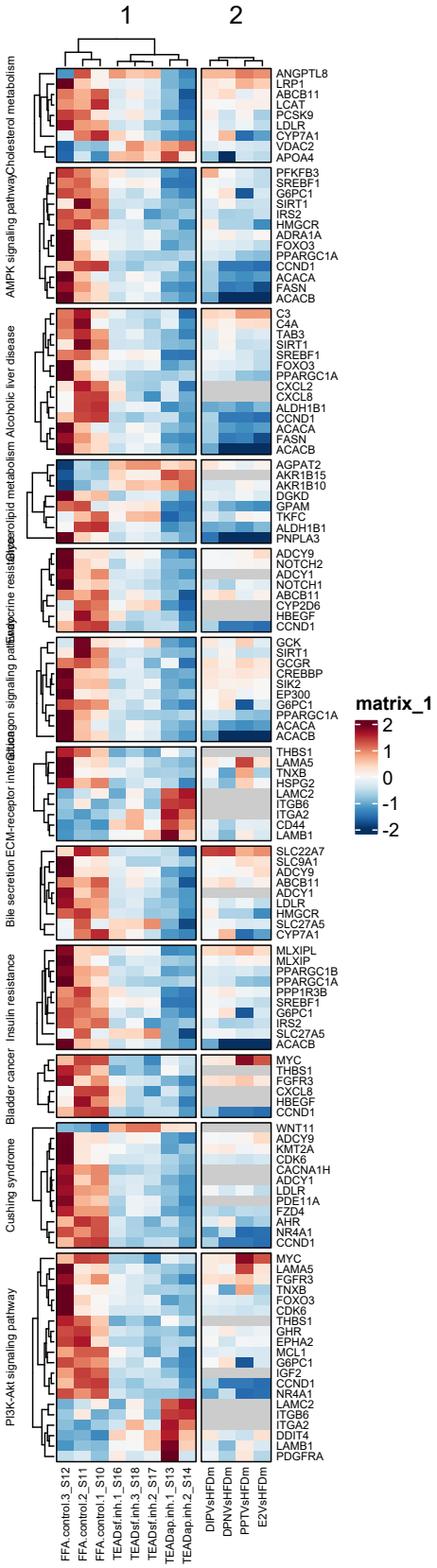
```
pval <- tailFraction(DEG_bs, non_DEG_bs$avg_bs, tail = 'right')

ggplot(non_DEG_bs, aes(x = avg_bs)) +
  geom_density(lwd = 1, colour = 'black',
               fill = '#616161', alpha = 1) +
  geom_vline(xintercept = DEG_bs, lwd = 1.5) +
  scale_x_continuous(limits = c(0,0.5),
                     breaks = c(0,0.25,0.5),
                     expand = expansion(mult = c(.02, .02))) +
  scale_y_continuous(expand = expansion(mult = c(.02, .1))) +
  annotate("text",  x=Inf, y = Inf, label = paste('P =',pval), vjust=1.5, hjust=1.2, size=5) +
  xlab('# of TEAD1 binding sites') +
  ylab('Density') +
  theme_bw(base_size = 12)
```



Expression heatmap top 12 KEGG pathways

```
top12_kegg <- readRDS('results/top12KEGGpathways_gene_content.rds')

df <- dplyr::bind_rows(top12_kegg) %>%
  dplyr::rename(external_gene_name = SYMBOL) %>%
  dplyr::inner_join(spheroid_tpm, by = 'external_gene_name') %>%
  dplyr::mutate(external_gene_name_unique = uniquify(external_gene_name, sep='_'))

mouse_human_orthologs <- mouse_human_orthologs %>%
  dplyr::filter(!GeneSymbol_human == '') %>%
  dplyr::filter(!GeneSymbol_mouse == '')

selected_orthologs <- df %>%
  dplyr::select(external_gene_name) %>%
  dplyr::filter(!duplicated(external_gene_name)) %>%
  dplyr::mutate(external_gene_name_mouse = dplyr::recode(external_gene_name,
                                                         !!!setNames(mouse_human_orthologs$GeneSymbol_mouse,
                                                                     mouse_human_orthologs$GeneSymbol_human)))
```

```r
mouse_ERtreatmentvsHFDm <- lapply(names(DEGs_mouse$unfilt[5:8]), function(x){
  DEGs_mouse$unfilt[[x]] %>%
    dplyr::select(external_gene_name, log2FoldChange) %>%
    dplyr::rename(!!x := log2FoldChange)
}) %>% dplyr::bind_cols() %>%
  dplyr::select(-paste('external_gene_name', seq(3,7,2), sep='...')) %>%
  dplyr::rename(external_gene_name = 'external_gene_name...1') %>%
  dplyr::filter(external_gene_name %in% selected_orthologs$external_gene_name_mouse) %>%
  dplyr::mutate(external_gene_name =dplyr::recode(external_gene_name,
                                                  !!!setNames(mouse_human_orthologs$GeneSymbol_human,
                                                              mouse_human_orthologs$GeneSymbol_mouse)))

df <- df %>%
  left_join(mouse_ERtreatmentvsHFDm, by = 'external_gene_name')

m <- df %>%
  tibble::column_to_rownames(var = 'external_gene_name_unique') %>%
  dplyr::select(FFA.control.1_S10:TEADsf.inh.3_S18) %>%
  scaleData(method = 'zscore') %>%
  dplyr::bind_cols(dplyr::select(df, DPNVsHFDm:PPTVsHFDm))


p <- Heatmap(m, width = unit(40, "mm"), height = unit(260, "mm"),
             row_split = factor(df$Pathway, levels = unique(df$Pathway)),
             column_split = c(rep(1,8), rep(2,4)),
             col = circlize::colorRamp2(breaks=seq(-2, 2, length.out=11),
                                        colors=colPals$RdBu),
             na_col = "grey80",
             row_title = unique(df$Pathway),
             row_title_gp = grid::gpar(fontsize = 6),
             row_labels = setNames(df$external_gene_name, df$external_gene_name_unique)[rownames(m)],
             row_names_gp = grid::gpar(fontsize = 6),
             column_names_gp = grid::gpar(fontsize = 6),
             cluster_row_slices = FALSE,
             cluster_column_slices = FALSE,
             border = T,
             gap = unit(1, 'mm'),
             row_dend_width = unit(5, 'mm'),
             column_dend_height = unit(5, 'mm'))

draw(p)
```


1
2
Cholesterol metabolism
ANGPTL8
LRP1
ABCB11
LCAT
PCSK9
LDLR
CYP7A1
VDAC2
APOA4
AMPK signaling pathway
PFKFB3
SREBF1
G6PC1
SIRT1
IRS2
HMGCR
ADRA1A
FOXO3
PPARGC1A
CCND1
ACACA
FASN
ACACB
Alcoholic liver disease
C3
C4A
TAB3
SIRT1
SREBF1
FOXO3
PPARGC1A
CXCL2
CXCL8
ALDH1B1
CCND1
ACACA
FASN
ACACB
Glycerolipid metabolism
AGPAT2
AKR1B15
AKR1B10
DGKD
GPAM
TKFC
ALDH1B1
PNPLA3
Glucagon signaling pathway
ADCY9
NOTCH2
ADCY1
NOTCH1
ABCB11
CYP2D6
HBEGF
CCND1
Glucagon signaling pathway
GCK
SIRT1
GCGR
CREBBP
SIK2
EP300
G6PC1
PPARGC1A
ACACA
ACACB
ECM-receptor interaction
THBS1
LAMA5
TNXB
HSPG2
LAMC2
ITGB6
ITGA2
CD44
LAMB1
Bile secretion
SLC22A7
SLC9A1
ADCY9
ABCB11
ADCY1
LDLR
HMGCR
SLC27A5
CYP7A1
Insulin resistance
MLXIPL
MLXIP
PPARGC1B
PPARGC1A
PPP1R3B
SREBF1
G6PC1
IRS2
SLC27A5
ACACB
Bladder cancer
MYC
THBS1
FGFR3
CXCL8
HBEGF
CCND1
Cushing syndrome
WNT11
ADCY9
KMT2A
CDK6
CACNA1H
ADCY1
LDLR
PDE11A
FZD4
AHR
NR4A1
CCND1
PI3K-Akt signaling pathway
MYC
LAMA5
FGFR3
TNXB
FOXO3
CDK6
THBS1
GHR
EPHA2
MCL1
G6PC1
IGF2
CCND1
NR4A1
LAMC2
ITGB6
ITGA2
DDIT4
LAMB1
PDGFRA
matrix_1
2
1
0
-1
-2
FFA.control.3_S12
FFA.control.2_S11
FFA.control.1_S10
TEADsf.inh.1_S16
TEADsf.inh.3_S18
TEADsf.inh.2_S17
TEADap.inh.1_S13
TEADap.inh.2_S14
DIPVsHFDm
DPNVsHFDm
PPTVsHFDm
E2VsHFDm

KEGG pathway genes changed by TEAD1i and ER activation

```
# entrez ID to ensembl ID matching
ann_genes <- getBM(
  mart=useDataset("hsapiens_gene_ensembl", useEnsembl(biomart = "ENSEMBL_MART_ENSEMBL", version = 109)),
  attributes=c("ensembl_gene_id", "entrezgene_id", "external_gene_name")) %>%
  filter(!is.na(entrezgene_id)) %>%
  filter(!external_gene_name == '')

# KEGG pathway genes
top12KEGG_all <- read.table(
  file = 'data/KEGG_pathways_april2023.txt',
  sep = '\t',
  header = TRUE,
  quote = '') %>%
  filter(KEGG_pathway_identifier %in% c("hsa04979", "hsa04152", "hsa04936", "hsa00561",
                                        "hsa01522", "hsa04922", "hsa04512", "hsa04976",
                                        "hsa04931", "hsa05219", "hsa04934", "hsa04151")) %>%
  filter(!duplicated(KEGG_ID)) %>%
  mutate(ensembl_gene_id = recode(KEGG_ID, !!!setNames(ann_genes$ensembl_gene_id,
                                                       ann_genes$entrezgene_id))) %>%
  mutate(GeneID_human = ensembl_gene_id) %>%
  inner_join(mouse_human_orthologs, by = 'GeneID_human') %>%
  filter(!(duplicated(GeneSymbol_human) | duplicated(GeneSymbol_human, fromLast = TRUE)))

top12_kegg <- readRDS('results/top12KEGGpathways_gene_content.rds')
top12_kegg_TEADap_inh_DEGs <- TEADap_inh_vs_control %>%
  filter(ensembl_gene_id %in% top12KEGG_all$ensembl_gene_id) %>%
  mutate(GeneID_human = ensembl_gene_id) %>%
  inner_join(mouse_human_orthologs, by = 'GeneID_human') %>%
  filter(!(duplicated(GeneSymbol_human) | duplicated(GeneSymbol_human, fromLast = TRUE)))

ER_activation_DEGs <- bind_rows(DEGs_mouse$filt[5:8]) %>%
  filter(!duplicated(ensembl_gene_id)) %>%
  mutate(GeneID_mouse = ensembl_gene_id) %>%
  inner_join(mouse_human_orthologs, by = 'GeneID_mouse') %>%
  filter(!(duplicated(GeneSymbol_mouse) | duplicated(GeneSymbol_mouse, fromLast = TRUE))) %>%
  filter(GeneID_human %in% top12KEGG_all$ensembl_gene_id)

top12KEGG_all_nonTEAD1i_DEGs <- top12KEGG_all %>%
  filter(!GeneSymbol_human %in% top12_kegg_TEADap_inh_DEGs$GeneSymbol_human)

# random_genes <- lapply(seq(1,1000), function(x) sample(x = top12KEGG_all_nonTEAD1i_DEGs$GeneSymbol_human, size = length(top12_kegg_TEADap_inh_DEGs
# saveRDS(random_genes, file = 'results/random_top12KEGG_nonTEADap_inh_sets.rds')
random_genes <- readRDS(file = 'results/random_top12KEGG_nonTEADap_inh_sets.rds')

# TEADap inhibitor DEG and non-DEG overlaps for genes in top 12 KEGG pathways that are are also
# changed by ER-agonist treatments in mouse HFD models
top12KEGG_nonTEADap_inh_DEG_ER_overlap <- lapply(random_genes, function(x){
  length(intersect(x, ER_activation_DEGs$GeneSymbol_human))/length(x)*100
}) %>% unlist() %>% as.data.frame() %>% dplyr::rename('perc_ER_activ' = '.')

top12KEGG_TEADap_inh_DEG_ER_overlap <- length(intersect(top12_kegg_TEADap_inh_DEGs$GeneSymbol_human, ER_activation_DEGs$GeneSymbol_human))/length(

print('Overlap for non-TEADap inhibitor DEG sets:')
```

```
## [1] "Overlap for non-TEADap inhibitor DEG sets:"
```

```
summary(top12KEGG_nonTEADap_inh_DEG_ER_overlap)
```

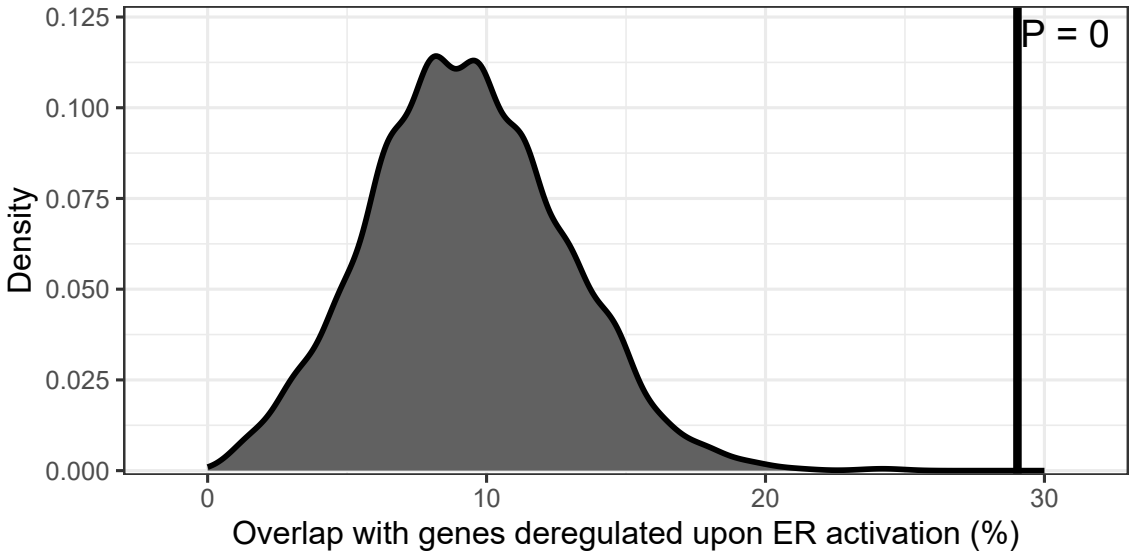
```
##  perc_ER_activ
##  Min.   : 1.613
##  1st Qu.: 6.452
##  Median : 9.677
##  Mean   : 9.305
##  3rd Qu.:11.290
##  Max.   :24.194
```

```
paste('Overlap for TEADap inhibitor DEG set:', top12KEGG_TEADap_inh_DEG_ER_overlap)
```

```
## [1] "Overlap for TEADap inhibitor DEG set: 29.0322580645161"
```

```
pval <- tailFraction(top12KEGG_TEADap_inh_DEG_ER_overlap,
                     top12KEGG_nonTEADap_inh_DEG_ER_overlap$perc_ER_activ,
                     tail = 'right')

ggplot(top12KEGG_nonTEADap_inh_DEG_ER_overlap, aes(x = perc_ER_activ)) +
  geom_density(lwd = 1, colour = 'black',
               fill = '#616161', alpha = 1) +
  geom_vline(xintercept = top12KEGG_TEADap_inh_DEG_ER_overlap, lwd = 1.5) +
  scale_x_continuous(limits = c(0,30),
```

```
                    breaks = c(0,10,20,30),
                    expand = expansion(mult = c(.1, .1))) +
  scale_y_continuous(expand = expansion(mult = c(.01, .12))) +
  annotate("text",  x=Inf, y = Inf, label = paste('P =',pval), vjust=1.5, hjust=1.2, size=5) +
  xlab('Overlap with genes deregulated upon ER activation (%)') +
  ylab('Density') +
  theme_bw(base_size = 12)
```



SessionInfo

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
##  [1] grid      tcltk     stats4    stats     graphics  grDevices utils
##  [8] datasets  methods   base
##
## other attached packages:
##  [1] ComplexHeatmap_2.14.0 GenomicRanges_1.50.1  GenomeInfoDb_1.34.9
##  [4] biomaRt_2.54.1        Mfuzz_2.58.0          DynDoc_1.76.0
##  [7] widgetTools_1.76.0    e1071_1.7-13          org.Hs.eg.db_3.16.0
## [10] AnnotationDbi_1.60.2  IRanges_2.32.0        S4Vectors_0.36.0
## [13] Biobase_2.58.0        BiocGenerics_0.44.0   lubridate_1.9.2
## [16] forcats_1.0.0         stringr_1.5.0         dplyr_1.1.0
## [19] purrr_1.0.1           readr_2.1.4           tidyr_1.3.0
## [22] tibble_3.2.1          ggplot2_3.4.2         tidyverse_2.0.0
## [25] clusterProfiler_4.6.2
##
## loaded via a namespace (and not attached):
##   [1] circlize_0.4.15        shadowtext_0.1.2       fastmatch_1.1-3
##   [4] BiocFileCache_2.6.1    plyr_1.8.8             igraph_1.3.5
##   [7] lazyeval_0.2.2         splines_4.2.1          BiocParallel_1.32.3
##  [10] digest_0.6.30          foreach_1.5.2          yulab.utils_0.0.6
##  [13] htmltools_0.5.5        GOSemSim_2.24.0        viridis_0.6.3
##  [16] GO.db_3.16.0           fansi_1.0.3            magrittr_2.0.3
##  [19] memoise_2.0.1          cluster_2.1.4          doParallel_1.0.17
##  [22] tzdb_0.4.0             Biostrings_2.66.0      graphlayouts_0.8.4
```

```
##  [25] matrixStats_0.63.0     timechange_0.2.0       enrichplot_1.18.4
##  [28] prettyunits_1.1.1      colorspace_2.0-3       blob_1.2.4
##  [31] rappdirs_0.3.3         ggrepel_0.9.2          xfun_0.39
##  [34] crayon_1.5.2           RCurl_1.98-1.9         jsonlite_1.7.2
##  [37] scatterpie_0.2.1       iterators_1.0.14       ape_5.6-2
##  [40] glue_1.6.2             polyclip_1.10-4        gtable_0.3.3
##  [43] zlibbioc_1.44.0        XVector_0.38.0         GetoptLong_1.0.5
##  [46] shape_1.4.6            scales_1.2.1           DOSE_3.24.2
##  [49] DBI_1.1.3              Rcpp_1.0.9             viridisLite_0.4.2
##  [52] progress_1.2.2         clue_0.3-63            gridGraphics_0.5-1
##  [55] tidytree_0.4.4         bit_4.0.5              proxy_0.4-27
##  [58] httr_1.4.6             fgsea_1.24.0           RColorBrewer_1.1-3
##  [61] pkgconfig_2.0.3        XML_3.99-0.12          farver_2.1.1
##  [64] dbplyr_2.3.3           utf8_1.2.2             labeling_0.4.2
##  [67] ggplotify_0.1.1        tidyselect_1.2.0       rlang_1.1.1
##  [70] tkWidgets_1.76.0       reshape2_1.4.4         munsell_0.5.0
##  [73] tools_4.2.1            cachem_1.0.6           downloader_0.4
##  [76] cli_3.4.1              generics_0.1.3         RSQLite_2.2.19
##  [79] gson_0.1.0             evaluate_0.21          fastmap_1.1.0
##  [82] yaml_2.3.7             ggtree_3.6.2           knitr_1.43
##  [85] bit64_4.0.5            tidygraph_1.2.2        KEGGREST_1.38.0
##  [88] ggraph_2.1.0           nlme_3.1-162           aplot_0.1.10
##  [91] xml2_1.3.5             compiler_4.2.1         rstudioapi_0.15.0
##  [94] filelock_1.0.2         curl_5.0.1             png_0.1-8
##  [97] treeio_1.22.0          tweenr_2.0.2           stringi_1.7.8
## [100] highr_0.10             lattice_0.20-41        Matrix_1.5-3
## [103] vctrs_0.6.2            pillar_1.9.0           lifecycle_1.0.3
## [106] GlobalOptions_0.1.2    data.table_1.14.6      cowplot_1.1.1
## [109] bitops_1.0-7           patchwork_1.1.2        qvalue_2.30.0
## [112] R6_2.5.1               gridExtra_2.3          codetools_0.2-19
## [115] MASS_7.3-60            rjson_0.2.21           withr_2.5.0
## [118] GenomeInfoDbData_1.2.9 parallel_4.2.1         hms_1.1.3
## [121] ggfun_0.1.1            HDO.db_0.99.1          class_7.3-22
## [124] rmarkdown_2.23         Cairo_1.6-0            ggforce_0.4.1
```